Birzeit University

Department of Electrical & Computer Engineering

First Semester, 2023/2024

ENCS5343 Computer Vision

Course Project + Assignment 3

Due Date January 28, 2024

1. Background

Handwriting recognition is a computer vision problem that involves a computer identifying handwritten script and transforming the text from sources such as documents or touchscreens into a machine-understandable format. The input image can be offline (from a piece of paper or a photograph) or online (from a digital source, such as touchscreens).

Handwritten text in each language has a wide range of patterns and styles, which are influenced by factors such as age, background, native language, and mental state. Various machine learning methods, such as K-nearest neighbors (KNNs), Support Vector Machines (SVMs), transfer learning, and deep learning techniques such as Neural Networks (NNs), have been extensively investigated in the field of automatic handwritten recognition. Convolutional Neural Networks (CNNs) have recently been used in the majority of studies.

Arabic writing is semi-cursive and written from right to left, with 28 characters in the alphabet. Because each character has multiple shapes depending on its position in the word, automatic handwritten recognition of Arabic script is more difficult than in other languages. Because of these factors, automatic handwritten recognition of Arabic script is more difficult than in other languages.

2. Objectives:

This assignment aims to introduce students to the challenges and techniques of Arabic Handwritten Character Recognition (AHCR) using Convolutional Neural Networks (CNNs). Here are some main objectives

- Building a Basic CNN for AHCR, which includes:
  - ✔ Network architecture: define a basic CNN architecture with convolutional layers, pooling layers, and activation functions.
  - ✔ Loss function and optimization: Introduce the concept of loss functions (e.g., cross-entropy) and optimizers (e.g., Adam) for training the CNN.
  - ✔ Training and evaluation: Guide students through training the model on a provided dataset and evaluating its performance using metrics like accuracy.
  - ✔ Visualization: Instruct students to visualize the CNN model's performance, such as the loss curve.

- Exploring advanced techniques, including:
  - ✔ Introduce data augmentation techniques: Discuss how techniques like random cropping, rotation, and elastic deformations can improve model generalizability.
  - ✔ Explore the use of pre-trained models: Consider incorporating pre-trained CNN models (e.g., VGG16) and fine-tuning them for AHCR.
  - ✔ Touch upon more advanced architectures: Briefly mention deeper and more complex CNN architectures like ResNet or DenseNet and their potential in AHCR.

3. Tasks:

The assignment contains four main tasks, defined as follows:

Task 1: Build and train a custom CNN network for AHCR. To build a custom CNN, you need to define the following:

- Architecture:
  - Number of layers: The depth of the network, typically starting with 1 to 2 convolutional layers and gradually increasing for more complex tasks.
  - Types of layers: convolutional layers, pooling layers (max, average, global), fully connected layers, and potentially other specialized layers (e.g., dropout, batch normalization).
  - Activation functions: ReLU (common choice), sigmoid, tanh, or others for specific needs.
- Convolutional Layer Parameters:
  - Number of filters: controls the number of feature maps extracted at each layer. More filters can capture more features but increase computational costs.
  - Filter size: determines the receptive field of the filters, often starting with 3x3 or 5x5 and increasing in deeper layers.
  - Stride: The step size of the filter movement affects the output size and computational complexity.
  - Padding: zero-padding input images to preserve spatial dimensions and capture edge features.
- Pooling Layer Parameters:
  - Pool size: The size of the pooling window, common choices being 2x2 or 3x3.
  - Pool type: max pooling or average pooling, each with different effects on feature preservation.
- Fully Connected Layer Parameters:
  - Number of neurons: related to the complexity of the task and the number of classes for classification.
- Training Hyperparameters:
  - Learning rate: controls how much the model's weights are updated during training.
  - Batch size: the number of samples processed per training step, affecting gradient updates and convergence speed.
  - Epochs: The number of times the model trains on the entire dataset.

- Optimizer: algorithm for updating model weights (e.g., Adam, SGD, RMSprop).
- Overfitting Handling: Techniques to prevent overfitting (e.g., dropout, L1/L2 regularization).

For this task, you can experiment with a set of architectures or you can refer to similar work (see the list of references below) and choose an initial architecture and then do hyperparameter tuning.

For each model you trained, you need to plot the following:

1. Training loss vs. epoch.
2. Validation loss vs. epoch.
3. Training accuracy vs. epoch.
4. Testing accuracy vs. epoch.

References:

- Alsayed, Alhag & Li, Chunlin & Ahamed, & Hazim, Mohammed & Obied, Zainab. (2023). Arabic Handwritten Character Recognition Using Convolutional Neural Networks. 10.21203/rs.3.rs-3141935/v1.
- Alwagdani, M.S.; Jaha, E.S. Deep Learning-Based Child Handwritten Arabic Character Recognition and Handwriting Discrimination. Sensors **2023**, 23, 6774. https://doi.org/10.3390/s23156774.
- Altwaijry, N., Al-Turaiki, I. Arabic handwriting recognition system using convolutional neural network. Neural Comput & Applic **33**, 2249–2261 (2021).
- Balaha, H.M., Ali, H.A., Youssef, E.K. et al. Recognizing arabic handwritten characters using deep learning and genetic algorithms. Multimed Tools Appl **80**, 32473–32509 (2021).
- Arabic Handwritten Character Recognition based on Convolution Neural Networks and Support Vector Machine. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 8, 2020
- Others…

Task 2: Retrain the network selected from **Task 1** after doing data augmentation.

Data augmentation is a powerful technique for enhancing the diversity and the size of your training data without the need for additional data collection. By applying various transformations to existing data points, you can train models that generalize better to unseen examples and improve their overall performance.

Data augmentation encompasses a wide range of techniques, only a subset of which are suitable for training the AHCR system. In this task, you must **select at least three data augmentation techniques that are appropriate** and for the model you trained, you need to plot the following:

1. Training loss vs. epoch.
2. Validation loss vs. epoch.
3. Training accuracy vs. epoch.
4. Testing accuracy vs. epoch.

Compare the results you obtained with the results of Task 1.

Task 3: Select a CNN network from a list of well-known and published CNN architectures, such as LeNet, AlexNet, ResNet, and so on. **You must make a tradeoff between accuracy and network complexity with respect to the problem and the dataset provided.** Train it using the data augmentation techniques you used in Task 2.

For the model you trained, you need to plot the following:

1. Training loss vs. epoch.
2. Validation loss vs. epoch.
3. Training accuracy vs. epoch.
4. Testing accuracy vs. epoch.

Compare the results you obtained with the results of Task 1 and Task 2.

Task 4: Use a pre-trained CNN network on **similar tasks** and choose the **appropriate transfer learning method** to fine tune the pretrained network on the given dataset.

For the model you trained, you need to plot the following:

1. Training loss vs. epoch.
2. Validation loss vs. epoch.
3. Training accuracy vs. epoch.
4. Testing accuracy vs. epoch.

Compare the results you obtained with the results of Task 1, Task 2, and Task 3.

4. Data set

Use the following dataset to train and test the models:

https://drive.google.com/file/d/1ZQ8fSD6WgkXFBKIxMRBMn0-gTwzFjUvz/view?usp=sharing

5. Report

Write a comprehensive report that includes the following:

1. **Introduction**
2. **Experimental setup and results**
   - Describe the evaluation methodology, including the datasets used, evaluation metrics, and experimental setup.
   - **Present, analyze, and discuss** the results of the evaluation, including accuracy and loss curves.
   - Analyze the effectiveness of using data augmentation, published CNN networks, and pretrained networks.
3. **Conclusion**
   - Summarize the key findings and achievements of the project.

6. Notes:

- Make sure your code is clean and well indented; variables have meaningful names, etc.
- Make sure your code has enough comments inserted to add clarity.
- Work in groups of at most two students
- Deadline: Sunday, January 28, 2024, at 11:59pm. Please submit your project (code and report) through Ritaj as a reply to this message.
- To turn in your assignment, Create a folder called [StudentID_FirstName] and place all of your solution's files in it, then compress it and submit only the.zip file. You can submit both a code (.py) and a short report (.pdf) that summarizes the results and their discussion for the solution files. You can also use Jupyter Notebook to prepare your solution in a single (.ipynb) file that includes both code cells and text cells that discuss the results. Do not submit links to your notebook if you use Colab. Only the.ipynb file is needed.
- This project is a group effort: instances of cheating will result in you failing the course.